

```
UUU      UUU VVV      VVV      111      RRRRRRRRRRRR      0000000000      MMM      MMM
UUU      UUU VVV      VVV      111      RRRRRRRRRRRR      0000000000      MMM      MMM
UUU      UUU VVV      VVV      111      RRRRRRRRRRRR      0000000000      MMM      MMM
UUU      UUU VVV      VVV      111111      RRR      RRR      000      000      MMMMMM      MMMMMM
UUU      UUU VVV      VVV      111111      RRR      RRR      000      000      MMMMMM      MMMMMM
UUU      UUU VVV      VVV      111111      RRR      RRR      000      000      MMMMMM      MMMMMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUU      UUU VVV      VVV      111      RRR      RRR      000      000      MMM      MMM      MMM
UUUUUUUUUUUUUUUUUU      VVV      1111111111      RRR      RRR      0000000000      MMM      MMM
UUUUUUUUUUUUUUUUUU      VVV      1111111111      RRR      RRR      0000000000      MMM      MMM
UUUUUUUUUUUUUUUUUU      VVV      1111111111      RRR      RRR      0000000000      MMM      MMM
```


(2)	70	DECLARATIONS
(3)	144	QRDY Bootstrap device initialization
(4)	373	QRDY Bootstrap device QIO
(5)	489	QRDY Bootstrap device disconnect


```
00000001 0000 1 BOOT_UV1_SWITCH = 1 ; Build Micro-VAX I bootstrap emulator
00000001 0000 2 PQ == 1
0000 1 .NLIST CND
0000 6 .TITLE PQBTDRIVR - QRDY BOOT DRIVER
0000 8 .IDENT 'V03-006'
0000 9
0000 10 :
0000 11 :*****
0000 12 :*
0000 13 :* COPYRIGHT (c) 1978, 1980, 1982 BY
0000 14 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 15 :* ALL RIGHTS RESERVED.
0000 16 :*
0000 17 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 18 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 19 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 20 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 21 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 22 :* TRANSFERRED.
0000 23 :*
0000 24 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 25 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 26 :* CORPORATION.
0000 27 :*
0000 28 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 29 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 30 :*
0000 31 :*
0000 32 :*****
0000 33 :
0000 34 :
0000 35 :++
0000 36 : FACILITY: BOOTS
0000 37 :
0000 38 : ABSTRACT:
0000 39 : This module contains the bootstrap device driver for the
0000 40 : UDA 50 disks.
0000 41 :
0000 42 : ENVIRONMENT: IPL 31, kernel mode, code must be PIC
0000 43 :
0000 44 : AUTHOR: Kerbey T. Altmann, CREATION DATE: 20-Nov-1981
0000 45 :
0000 46 : MODIFIED BY:
0000 47 :
0000 48 : V03-006 KDM0073 Kathleen D. Morse 23-Aug-1983
0000 49 : Added $BQODEF for use by new version of TIMEDWAIT macro.
0000 50 :
0000 51 : V03-005 KDM0072 Kathleen D. Morse 18-Aug-1983
0000 52 : Conditionalize assembly to build a driver for the
0000 53 : QRDY controler.
0000 54 :
0000 55 : V03-004 KDM0059 Kathleen D. Morse 13-Jul-1983
0000 56 : Replace time-wait loops that use IPR TODR with the
0000 57 : new TIMEDWAIT macro.
0000 58 :
0000 59 : V03-003 KTA3064 Kerbey T. Altmann 03-Jul-1983
0000 60 : Fix page boundary problem.
```


PQBTDRIVR
V03-006

- QRDY BOOT DRIVER

E 12

10-AUG-1984 18:04:32 VAX/VMS Macro V04-00
9-JUL-1984 11:51:49 PUBTDRIVR.MAR;1

Page 2
(1)

0000 61 :
0000 62 :
0000 63 :
0000 64 :
0000 65 :
0000 66 :
0000 67 :
0000 68 :--

V03-002 KTA3059 Kerbey T. Altmann 23-Jun-1983
Add support for the boot device name.
V03-001 KTA3007 Kerbey T. Altmann 09-Oct-1982
Fix problem with setting VMB\$V_SCS.

QVS
V1.


```
0000 70 .SBTTL DECLARATIONS
0000 71 :
0000 72 : INCLUDE FILES:
0000 73 :
0000 74 :
0000 75 $BQODEF ; Boot qio offset definitions
0000 76 $BTODEF ; Boot device types
0000 77 $IODEF ; I/O function codes
0000 78 $MSCPDEF ; MSCP definitions
0000 79 $PRDEF ; Processor registers
0000 80 $PTEDEF ; Page table entries
0000 81 $RPBDEF ; RPB offsets
0000 82 $SSDEF ; Status codes
0000 83 $UBADEF ; UBA definitions
0000 84 $SUBIDEF ; 11/750 UBA definitions
0000 85 $VADEF ; Virtual addresses
0000 86 $VMBARGDEF ; VMB argument list offsets
0000 87 :
0000 88 :
0000 89 : EQUATED SYMBOLS:
0000 90 :
0000 91 :
00000000 0000 92 UDAIP = 0
00000002 0000 93 UDASA = 2
00000001 0000 94 GO = 1
00008000 0000 95 OWN = 1@15
0000000B 0000 96 S1 = 11
0000000E 0000 97 S4 = 14
0000 101 :
0000 102 :
0000 103 : OWN STORAGE:
0000 104 :
0000 105 :
0000 106 :
0000 107 : Boot driver table entry
0000 108 :
0000 109 :
0000 122 $BOOT_DRIVER DEVTYPE = BTDSK,UDA,- ; Device type (QRDX)
0000 123 SIZE = UD_DRVSIZE,- ; Driver size
0000 124 ADDR = START,- ; Driver starting address
0000 125 ENTRY = UD_DRIVER,- ; Driver entry point
0000 126 UNIT_INIT = UD_INIT,- ; Driver unit init entry
0000 127 UNIT_DISC = UD_DISC,- ; Driver disconnect entry
0000 128 DRVNAME = DSRDRVNAME,- ; Driver disk name
0000 129 AUXDRVNAME = PRTDRVNAME,- ; Driver port name
0000 130 DEVNAME = DEVNAME ; Boot device name
0000 131 :
0000 132 :
0000 133 START:
0000 134 DSKDRVNAME:
58 45 2E 52 45 56 49 52 44 55 44 00' 0000 135 .ASCIC /DUDRIVER.EXE/ ; Disk class driver filename
45 000C
0C 0000
000D 136 PRTDRVNAME:
58 45 2E 52 45 56 49 52 44 55 50 00' 000D 137 .ASCIC /PUDRIVER.EXE/ ; Port driver filename
45 0019
0C 000D
55 44 001A 138 DEVNAME:.ASCII /DU/ ; Boot device name
```



```
001C 144 .SBTTL QRDY Bootstrap device initialization
001C 146
001C 147 :++
001C 148 :
001C 149 : Inputs:
001C 150 :
001C 151 : R9 --> RPB
001C 152 : AP --> VMB argument list
001C 153 :
001C 154 : Outputs:
001C 155 :
001C 156 : R0 - status code
001C 157 :
001C 158 :--
001C 159 :
001C 160 UD_INIT:
001C 161 .ENABLE LSB
01FC 001C 162 .WORD ^M<R2,R3,R4,R5,R6,R7,R8>
001E 163
50 38 DB 001E 164 MFPR #PR$ MAPEN, R0 ; Get the mapping status
0B 50 E8 0021 165 BLBS R0,10$ ; If virtual, skip some set up
0024 166
0024 167 : Set up the SYSTEMID for the local UDA.
0024 168 :
51 34 A9 D0 0024 169 MOVL RPB$L IOVEC(R9),R1 ; Point to iovec
00'A1 94 0028 170 CLRB B^<BOO$GB UMR DP-BOO$AL_VECTOR>(R1) ; Set for Direct Data Path
2C AC 01 D0 002B 171 ASSUME VMB$V LOAD_SCS EQ 0
002B 172 MOVL #1,VMB$L_FLAGS(AP) ; Set a flag to load SCS code
002F 180
002F 181 :
002F 182 : Set up an interrupt vector.
002F 183 :
1E A9 01FC 8F B0 002F 184 10$: MOVW #<127*4>,RPB$W ROUBVEC(R9) ; Use the highest possible
57 54 A940 D0 0035 210 MOVL RPB$L CSRPHY(R9)[R0],R7 ; Get correct address of device CSR
52 0150'CF 9E 003A 211 MOVAB W^INTTBL,R2 ; Get the address of the init table
02 A2 52 FF000000 8F CB 003F 212 BICL3 #^C<^XFFFFFF>,R2,2(R2) ; Start address of init table into tbl
0048 213 ; transfer only 24 bits
12 50 E9 0048 214 BLBC R0,15$ ; If clr, then physical address
51 52 15 09 EF 004B 215 EXTZV #VAV$ VPN,#VASS VPN,R2,R1 ; Convert VA to virtual page number
51 50 B941 D0 0050 216 MOVL @RPB$C SVASPT(R9)[R1],R1 ; Get the physical page number
02 A2 15 09 51 F0 0055 217 INSV R1,#VAV$ VPN,#VASS VPN,2(R2) ; Set physical address of init tbl
0F 11 005B 218 BRB 20$
24 AC 57 FFFC0000 8F CB 005D 219 15$: BICL3 #^C<^X3FFFF>,R7,- ; Low 18 bits is CSR of boot device
0066 220 VMB$B SYSTEMID(AP)
28 AC 8000 8F B0 0066 221 MOVW #^x3000,VMB$B SYSTEMID+4(AP);
02 A2 10' C0 006C 222 20$: ADDL2 S^#<RING-INTTBL>,2(R2) ; Finally get real phys adr of ring
58 02 A7 B0 0070 223 MOVW UDASA(R7),R8 ; Controller online?
06 12 0074 224 BNEQ 21$ ; Br if no, initialize controller
00BE 30 0076 225 BSBW SET_CONTROLLER_CHARS ; Set controller characteristics
71 50 E8 0079 226 BLBS R0,40$ ; continue if success
007C 227 21$: ; If failure, re init once
007C 229 :
007C 230 : Now go thru the ridiculously complicated startup sequence. This is a
007C 231 : fugue in four parts.
007C 232 :
58 02 D0 007C 233 MOVL #2,R8 ; Make two tries at this
0150'CF 9E 007F 234 RETRY: MOVAB W^INTTBL, R3
```



```
52 0B D0 0084 235      MOVL    #S1, R2          ; Step flag
    67 B4 0087 236      CLRW    UDAIP(R7)        ; Poke the controller's CSR
    0089 237      ;
    0089 238      ; Wait 10 seconds
    0089 239      ;
    0089 240      TIME:    TIMEDWAIT TIME=#1000*1000, - ; Wait for 10 seconds
    0089 241      INS1=<MOVW    UDASA(R7), R4>, - ; Check the status register
    0089 242      INS2=<BLSS    25$, - ; Bit 15 set is the error indicator
    0089 243      INS3=<BBS     R2, R4, 25$>, - ; Done with this step?
    0089 244      DONELBL=25$ ; Label for exiting wait loop
    0D 50 E8 00BC 245      BLBS    R0, 26$        ; Br if not timed out
    00BF 246      ;
50  BD 58 F5 00BF 247      ERROR:  SOBGTR R8, RETRY    ; Try once again
    20D4 8F 3C 00C2 249      MOVZWL #SS$_DEVINACT, R0 ; Set failure status
    04 00C7 253      RET
    00C8 254      ;
    80000000 00C8 255      VALID:  .LONG    ^X80000000 ; Sign bit set
    00CC 256      ;
    54 B5 00CC 257      26$:      TSTW    R4          ; Check status register for error
    EF 19 00CE 258      BLSS    ERROR        ; Br if error
    00D0 259      ;
    02 A7 83 B0 00D0 260      30$:      MOVW    (R3)+, UDASA(R7) ; Send the controller the next step
    B1 52 0E F3 00D4 261      AOBLEQ    #S4, R2, TIME ; Set for next step
    00D8 262      ;
    00D8 263      ; Initialization complete. Write the packet address in the ring.
    00D8 264      ; Writing the addresses must be deferred until this point because the
    00D8 265      ; hardware zeroes the entire ring as a memory write check.
    00D8 266      ;
    00D8 267      ;
    00000040'8F C1 00D8 268      ADDL3    #<RSPPKT-RING>, - ; Response packet
    60'AF 52'AF 00DE 269      B^INTTBL+2, B^RD
    64'AF 52'AF 0C' C1 00E2 270      ADDL3    S^#<CMDPKT-RING>, - ; Command packet
    00E4 271      B^INTTBL+2, B^CD
    00E8 272      ;
    4D 10 00E8 274      BSBB    SET_CONTROLLER_CHARS ; Set controller characteristics
    27 50 E9 00EA 275      BLBC    R0, 50$        ; Exit if error
    00ED 276      ;
    00ED 277      ; Now bring the device on-line
    00ED 278      ;
    00ED 279      ;
    00ED 280      ;
    65 01 D0 00ED 281      40$:      MOVL    #1, (R5) ; Set command ref number
    04 A5 64 A9 9A 00F0 282      MOVZBL    RPB$W_UNIT(R9), 4(R5) ; Put unit number in cmd packet field
    08 A5 09 9A 00F5 283      MOVZBL    #MSCP$K_OP_ONLIN, 8(R5) ; Set opcode to bring drive online
52  00989680 8F D0 00F9 284      MOVL    #<10000*1000>, R2 ; Set large wait time 100 secs
    00F2 30 0100 285      BSBW    IO          ; Send it out
    OE 50 E9 0103 286      BLBC    R0, 50$        ; Br if failure
    50 08 88 0106 287      BISB    #8, R0        ; Fixup success to reflect media type
    OA 01AE'CF 07 E1 0109 288      BBC     #MSCP$V_UF_RMVBL, W^RSPPKT+MSCP$W_UNT_FLGS, 55$
    50 10 88 010F 289      BISB    #16, R0       ; Fixup success to reflect removable
    05 11 0112 290      BRB     55$
    50 0084 8F 3C 0114 291      50$:      MOVZWL    #SS$_DEVOFFLINE, R0 ; Change status to unit offline
    0119 292      55$:
    16 EF 0119 309      EXTZV    #MSCP$V_MTYP_D1, -
    05 011B 310      #MSCP$S_MTYP_D1, -
    01BC'CF 011C 311      W^RSPPKT+ -
```



```

51 51 08 78 011F 312      MSCPSL MEDIA_ID,R1      ; Pull out 2nd device character
1B EF 0120 313      ASHL #8,R1,RT      ; Stick it in high byte
05 01BC'CF 0124 314      EXTZV #MSCPSV_MTYD_D0,-
52 0126 315      #MSCPS$S_MTYD_D0,-
51 4040 8F A8 012A 316      W^RSPPKT+ -
FEE4 CF 52 51 A9 012B 317      MSCPSL MEDIA_ID,R2      ; Pull out 1st device character
04 0128 318      BISW #^X4040,R1      ; Make ASCII characters
0130 319      BISW3 R1,R2,DEVNAME      ; Set into driver name
0136 320      RET
0137 321
0137 322      ;
0137 323      ; small routine to use so that the controller characteristics can be set
0137 324      ;
0137 325      ;
0137 326      ;
0137 327      SET_CONTROLLER_CHARS:
0137 328
55 016C'CF 9E 0137 329      MOVAB W^CMDPKT,r5      ; Get the address of command packet
65 01 013C 330      MOVL #1,(R5)      ; Set command ref number
04 A5 D4 013F 331      CLRL 4(R5)      ; No unit
08 A5 04 9A 0142 332      MOVZBL #MSCPSK_OP_STCON,8(R5) ; Set opcode to set controller chars
0C A5 7C 0146 333      CLRQ 12(R5)      ; Clear data area
14 A5 7C 0149 334      CLRQ 20(R5)      ; To stop timeouts
009F 31 014C 335      BRW IO_STANDARD_WAIT      ; Send it out and return
014F 336
014F 337      .DISABLE LSB
00000150 014F 338      .=<. +1>8-2
0150 339
0150 340      ;
0150 341      ; RINGS
0150 342      ;
0150 343      ;
8000 0150 344      INTTBL: .WORD 0WN      ; Step 1 pattern
00000000 0152 345      .LONG 0      ; Step 2 & 3 pattern
0001 0156 346      .WORD 60
0158 347
0000 0000 0158 348      .WORD 0,0      ; Reserved
0000 015C 349      CMDINT: .WORD 0      ; Command status word
0000 015E 350      RSPINT: .WORD 0      ; Response status word
0160 351      RING:
00000000 0160 357      RD: .LONG 0      ; QBUS address of response ring
00000000 0164 358      CD: .LONG 0      ; QBUS address of command ring
0168 360      ;
0030 0168 361      .WORD 48      ; Length of message
0001 016A 362      .WORD 1      ; ID
0001 016C 363      CMDPKT: .WORD 1
0000019C 016E 364      .BLKW 23      ; Full envelope
019C 365      ;
000001A0 019C 366      .BLKW 2
000001D0 01A0 367      RSPPKT: .BLKW 24

```



```

01D0 373 .SBTTL QRDY Bootstrap device Q10
01D0 375
01D0 376 :++
01D0 377
01D0 378 Inputs:
01D0 379
01D0 380 R3 - base address of adapter's register space
01D0 381 R5 - lbn for current piece of transfer
01D0 382 R6 - contains 0
01D0 383 R7 - address of the device's CSR
01D0 384 R8 - size of transfer in bytes
01D0 385 R9 - address of the RPB
01D0 391 R10 - starting address of transfer
01D0 392 ** R2 - PTE as address for VA in R10
01D0 394 R11 - LBN at start of transfer
01D0 395
01D0 396 FUNC(AP)- I/O operation (IOS_READBLK or IOS_WRITEBLK only)
01D0 397 SIZE(AP)- Size of transfer in bytes
01D0 398 MODE(AP)- Address interpretation mode (0 = physical, 1 = virtual)
01D0 399
01D0 400 Implicit inputs:
01D0 401
01D0 402 RPB$W_UNIT - RPB field containing boot device unit number
01D0 403
01D0 404 Outputs:
01D0 405
01D0 406 R0 - status code
01D0 407
01D0 408 SSS_NORMAL - successful transfer
01D0 409 SSS_NOSUCHDEV - unsupported device
01D0 410 SSS_CTRLERR - fatal controller error
01D0 411
01D0 412 R3 - must be preserved
01D0 413
01D0 414
01D0 415 :--
01D0 416
00000010 01D0 417 FUNC = 16
00000014 01D0 418 MODE = 20
01D0 419
01D0 420 UD_DRIVER: ; UDA50 device driver.
01D0 421
01D0 422
01D0 423 : Translate the I/O function code into a device-dependent function
01D0 424 : code for this disk.
01D0 425
01D0 426
A0 AF 21 90 01D0 427 MOVB #MSCP$K_OP_READ, - ; Assume read
01D4 428 CNDPKT+MSCP$B_OPCODE
20 10 AC D1 01D4 429 CMPL FUNC(AP),#IOS_WRITEBLK ; Check for write function
04 12 01D8 430 BNEQ 20$ ; No, do read
96 AF 22 90 01DA 431 MOVB #MSCP$K_OP_WRITE, - ; Set write function code
01DE 432 CNDPKT+MSCP$B_OPCODE
A6 AF 55 D0 01DE 433 20$: MOVL R5,CNDPKT+MSCP$L_LBN ; Set the logical block number
92 AF 58 D0 01E2 434 MOVL R8,CNDPKT+MSCP$L_BYTE_CNT ; Set the byte count
92 AF 5A D0 01E6 439 MOVL R10,CNDPKT+MSCP$B_BUFFER ; Set the starting adr of xfer
92 AF 52 D0 01EA 440 MOVL R2,CNDPKT+MSCP$B_BUFFER+4 ; Set the PTE if R10 is VA

```



```

01EE 441
01EE 443 IO_STANDARD WAIT:
01EE 444      MOVCL #<1000*1000>,R2      ; Set minimum wait time
01F5 445 IO:      MOVW UDASA(R7),R4      ; Controller offline?
01F9 446      BNEQ ERROR1              ; Yep, error out
01FB 447      BLSW #OWN, CD+2           ; Set controller ownership
0202 448      BLSW #OWN, RD+2           ; Ditto
0209 449      MOVW UDAIP(R7),R4        ; Tell controller to read
020C 450      MOVW UDASA(R7),R4        ; Any problems?
0210 451      BNEQ ERROR1              ; Yes
0212 452      TIMEDWAIT TIME=R2,-      ; loop until ready
0212 453      INS1=<TSTW RD+2>,-      ;
0212 454      INS2=<BGEQ 30$>,-      ;
0212 455      DONELBL=30$
023D 456      BLBC R0,ERROR1           ; Br if timed out
0240 457      BLSW #OWN, RD+2           ; No, set controller ownship
0247 459      EXTZV #MSCP$V_ST_MASK,#MSCP$S_ST_MASK,RSPPKT+MSCP$W_STATUS,R0 ; Any drive errors?
024E 460      BEQL 10$                 ; Br if no
024E 461      CMPL #1,R0               ; Invalid command?
0250 462      BNEQ ERROR1              ; Br if no, ignore invalid command
0253 463      ; which is caused by multiple onlines
0255 464
0255 465 10$:
0255 466
0255 467
0255 468 ; Transfer is complete. Return with success status code.
0255 469
0255 470
0255 471
0255 472
0255 473
0255 474      MOVZWL #SS$_NORMAL,R0      ; SET COMPLETION CODE
0258 475      RSB                      ; AND RETURN
0259 476
0259 477
0259 478 ; Error occurred during transfer. Return and retry.
0259 479
0259 480
0259 481 ERROR1:
0259 482      MOVZWL #SS$_CTRLERR,R0      ; Set failure status
025E 483      RSB                      ; Return to BOOTDRIVR
```



```
025F 489      .SBTTL QRDY Bootstrap device disconnect
025F 491
025F 492 :++
025F 493 : This routine disconnect the boot device after a bugcheck dump.
025F 494 : It sends an AVAIL packet to the controller, in effect doing a
025F 495 : dismount of the system device. It is designed to be called
025F 496 : only from BUGCHECK immediately after the dump has finished.
025F 497 : It assumes virtual mapping turned on.
025F 498
025F 499 : Inputs:
025F 500
025F 501 :      R9 --> RPB
025F 502
025F 503 : Outputs:
025F 504
025F 505 :      R0 - status code
025F 506
025F 507 :--
025F 508
025F 509 UD_DISC:
025F 510      .ENABLE LSB
0090 025F 511      .WORD  ^M<R4,R7>
0261 512
0261 513      MOVZBL #MSCPSK OP_AVAIL,-      ; Make drive AVAILable
0263 514      CNDPKT+MSCPSB_OPCODE
0266 515      MOVL  RPB$CSRVR(R9),R7      ; Get correct address of device CSR
57  FFOE CF  A9  D0 026A 516      BSBB  IO_STANDARD_WAIT      ; Send it out
    82  10 026C 517      RET
    04 026D 518
0000026D 026D 519 UD_DRVSIZ=-START
026D 520
026D 521      .END
```

PQBTDRIVR
Symbol table

- QRDZ BOOT DRIVER

M 12

10-AUG-1984 18:04:32 VAX/VMS Macro V04-00
9-JUL-1984 11:51:49 PQBTDRIVR.MAR;1

Page 10
(5)

\$TABLE	= 00000000	R	02
BOOSAL_VECTOR	*****	X	03
BOOSGB_UMR_DP	*****	X	03
BOOT_UV1_SWITCH	= 00000001		
BQOSL_TENUSEC	= 0000003E		
BQOSL_UBDELAY	= 00000042		
BTD\$K_UDA	= 00000011		
CD	00000164	R	03
CMDINT	0000015C	R	03
CMDPKT	0000016C	R	03
DEVNAME	0000001A	R	03
DSKDRVNAME	00000000	R	03
ERROR	000000BF	R	03
ERROR1	00000259	R	03
FUNC	= 00000010		
GO	= 00000001		
INTTBL	00000150	R	03
IO	000001F5	R	03
IOS\$ WRITELBLK	= 00000020		
IO\$ STANDARD_WAIT	000001EE	R	03
MODE	= 00000014		
MSCP\$B_BUFFER	= 00000010		
MSCP\$B_OPCODE	= 00000008		
MSCP\$K_OP_AVAIL	= 00000008		
MSCP\$K_OP_ONLIN	= 00000009		
MSCP\$K_OP_READ	= 00000021		
MSCP\$K_OP_STCON	= 00000004		
MSCP\$K_OP_WRITE	= 00000022		
MSCP\$K_BYTE_CNT	= 0000000C		
MSCP\$K_LBN	= 0000001C		
MSCP\$K_MEDIA_ID	= 0000001C		
MSCP\$S_MTYPE_D0	= 00000005		
MSCP\$S_MTYPE_D1	= 00000005		
MSCP\$S_ST_MASK	= 00000005		
MSCP\$V_MTYPE_D0	= 0000001B		
MSCP\$V_MTYPE_D1	= 00000016		
MSCP\$V_ST_MASK	= 00000000		
MSCP\$V_UF_RMVBL	= 00000007		
MSCP\$W_STATUS	= 0000000A		
MSCP\$W_UNT_FLGS	= 0000000E		
OWN	= 00008000		
PQ	= 00000001	G	
PR\$ MAPEN	= 00000038		
PRTDRVNAME	0000000D	R	03
RD	00000160	R	03
RETRY	0000007F	R	03
RING	00000160	R	03
RPBSL_CSRPHY	= 00000054		
RPBSL_CSRVIR	= 00000058		
RPBSL_IOVEC	= 00000034		
RPBSL_SVASPT	= 00000050		
RPBSW_ROUBVEC	= 0000001E		
RPBSW_UNIT	= 00000064		
RSPINT	0000015E	R	03
RSPPKT	000001A0	R	03
S1	= 0000000B		
S4	= 0000000E		

SET_CONTROLLER_CHARS	00000137	R	03
SS\$-CTRLERR	= 00000054		
SS\$-DEVINACT	= 000020D4		
SS\$-DEVOFFLINE	= 00000084		
SS\$-NORMAL	= 00000001		
START	00000000	R	03
TIME	00000089	R	03
UDAIP	= 00000000		
UDASA	= 00000002		
UD_DISC	0000025F	R	03
UD_DRIVER	000001D0	R	03
UD_DRVSIZ	= 0000026D		
UD_INIT	0000001C	R	03
VAS\$-VPN	= 00000015		
VASV-VPN	= 00000009		
VALID	000000C8	R	03
VMBSB_SYSTEMID	00000024		
VMBS\$-ARGBYTCNT	0000003C		
VMBSL-CI_HIPFN	00000030		
VMBSL-FLAGS	0000002C		
VMBSL-HI_PFN	00000010		
VMBSL-LO_PFN	0000000C		
VMBSQ-FILECACHE	00000004		
VMBSQ-NODENAME	00000034		
VMBSQ-PFNMAP	00000014		
VMBSQ-UCODE	0000001C		
VMBSV-LOAD_SCS	= 00000000		

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	0000003C (60.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
BOOTDRIVR_4	00000028 (40.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
BOOTDRIVR_2	0000026D (621.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	14	00:00:00.07	00:00:00.24
Command processing	70	00:00:00.57	00:00:01.94
Pass 1	364	00:00:14.70	00:00:18.69
Symbol table sort	0	00:00:02.49	00:00:02.72
Pass 2	72	00:00:02.75	00:00:03.72
Symbol table output	10	00:00:00.11	00:00:00.13
Psect synopsis output	3	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	533	00:00:20.71	00:00:27.49

The working set limit was 1350 pages.

86032 bytes (169 pages) of virtual memory were used to buffer the intermediate code.

There were 90 pages of symbol table space allocated to hold 1567 non-local and 17 local symbols.

523 source lines were read in Pass 1, producing 16 object records in Pass 2.

24 pages of virtual memory were used to define 21 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
DISK\$STARWORK03:[GAMACHE.UV1ROM.VMS]LIBUV1.ML	7
DISK\$STARWORK03:[GAMACHE.UV1ROM.OBJ]VMB.MLB;3	4
SYS\$SYSROOT:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	18

1717 GETS were required to define 18 macros.

There were no errors, warnings or information messages.

MAC/LIS=LIS\$:PQBTDRIVR/OBJ=OBJ\$:PQBTDRIVR VMSS:BOOUV1SWT+VMSS:PUBTDRIVR+OBJ\$:VMB/LIB+VMSS:LIBUV1/LIB

0430 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

